

DOCUMENT RESUME

ED 240 979

IR 010 971

AUTHOR Wappler, Reinhold D.
TITLE Apple (LCSI) LOGO vs. MIT (Terrapin/Krell) LOGO: A Comparison for Grades 2 thru 4.
PUB DATE May 83
NOTE 10p.; Best copy available.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Comparative Analysis; Elementary Education; *Evaluation Criteria; Grade 2; Grade 3; Grade 4; Guides; Information Storage; Man Machine Systems; Media Selection; *Microcomputers; Programing; *Programing Languages; Screens (Displays); Vocabulary
IDENTIFIERS Debugging (Computers); *LOGO Programing Language

ABSTRACT

Two LOGO dialects are compared for appropriateness for use with second, third, and fourth grade students on the basis of 18 months of experience with teaching LOGO programing language at this level in a four-machine laboratory setting. Benefits and drawbacks of the dialects are evaluated in the areas of editing, screen modes, debugging, vocabulary, instruction manuals, typing and syntax simplicity, automatic retyping, speed of drawing, ability to save and load binary picture files, remark facility, and workspace management. Specifically discussed are the MIT 'TRACE' and 'SAVEPICT/READPICT' commands and the Apple 'PACKAGE' and 'BUY' commands. The Apple and MIT LOGO programs are then rated using the criteria named above and a factor weighting system. The MIT Program is more highly rated overall, though the Apple program is rated higher on screen modes, extent of primitive vocabulary, the 'BURY' command, automatic retype (for CONTROL Y), and speed of drawing. It is noted that the final decision, based on the weighting and intuition of the researchers, was an extremely close one. An update from July 2, 1983 describes procedures for an 'ED ALL' capability to edit entire workspace in Apple LOGO. An update of February 1984 incorporates new experience with a second version of Terrapin LOGO. (LMM)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

* This document has been reproduced as
received from the person or organization
originating it.

□ Minor changes have been made to improve
reproduction quality.

• Points of view or opinions stated in this docu-
ment do not necessarily represent official NIE
position or policy.

APPLE (LCSI) LOGO
VS.

MIT (TERRAPIN/KRELL) LOGO
A COMPARISON FOR GRADES 2 THRU 4

REINHOLD D. WAPPLER
NEW CANAAN COUNTRY SCHOOL
NEW CANAAN, CONNECTICUT
MAY, 1983

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

W.D. Wappler

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

We have accumulated 18 months of experience with 2nd, 3rd, and 4th grades teaching with LOGO in a four machine laboratory setting. We used first LCSI LOGO during the summer and the first half of this past school year, and subsequently switched to MIT LOGO with a new group of children for the latter half year. Missing some of the excellent and unique features of the LCSI (Logo Computer Systems, Inc.) APPLE LOGO version, we had decided to switch back to that version commencing next September, 1983. However, after writing down our analysis of the benefits and drawbacks of both dialects, reconsidering the weight assigned to each of our criteria, and a brief refresher in LCSI, we decided to continue with the MIT version. It was an extremely close decision. Had the sequence of use been reversed, we might have decided to keep on with the familiarity of our most recent experience. We might well reconsider in 1984. Other groups might well come to the opposite conclusion based on different weighting of other or additional criteria. Our reasons and analysis follow.

EDITING:

MIT allows editing either of one procedure or of all procedures. You cannot call up two or three related procedures for editing as is possible with APPLE, e.g. by typing 'EDIT [PROC1 PROC2 PROC3]'. We had thought on the other hand that there was no command 'EDIT ALL' in APPLE, that you could not conveniently edit and view the entire workspace. This deficiency in APPLE LOGO has been addressed with a set of procedures published in the April 1983 issue of TURTLE NEWS (reproduced on an attachment hereto with a discussion of limitations).

In APPLE, the commands ED, ERase, and PD (printout) must be followed by a quoted procedure name, a nuisance for young typists. Contrast this with MIT's no requirement for quotes following any of these frequently used commands, plus the fact that a child need type only ED (Return) to edit the procedure last edited. This allows a student programmer quickly to

MIT TERRAPIN/KRELL VS APPLE LCSII LOGOS

alternate between editor and command level, usually to iterate his or her way to a number or command that works. We have furthermore not found the APPLE "mini-editor", entered by typing TO, to be of much use since its editing capabilities are limited to the current line only. CONCLUSION: Significant advantages in editing and typing ease to MIT.

SCREEN MODES:

In addition to NOWRAP (never used) and WRAP/FENCE modes in MIT and APPLE respectively, there is a WINDOW mode in APPLE. In WRAP, long lines drawn consequent to big number inputs exit one side of the screen and reappear on the other, continuing to "wrap" around in this manner until the entire line length is drawn. Unless carefully controlled, this results almost all the time in messy screens filled with meaningless lines. Students take a persistent (and fruitless) delight in filling their screens this way. We are tempted to discourage this behaviour, but of course children love big numbers, and should experiment with them. This has led us to place significant value on APPLE's WINDOW mode. In WINDOW mode the screen is simply a window on a much larger Cartesian grid, over 100000 turtle steps high and wide. SQUARE with side 50000 can now become a real and coherent experience to a child. There are also design and magnification opportunities in Poly's that encourage and make meaningful the use of large numbers. CONCLUSION: WINDOW mode is a significant advantage for APPLE.

DEBUGGING-- THE MIT 'TRACE' COMMAND:

For student debugging of what are to them complex or tricky procedures, a TRACE command to run procedures one line at a time can be of significant value, and MIT has it as a primitive, immediately and conveniently available. In APPLE there is a 'STEPPER' procedure which you have to write, and which you could include among the "buried" (see below) startup procedures. Nevertheless, it is far more awkward in APPLE and a built-in, simple TRACE capability is missed, particularly by the more advanced students. CONCLUSION: TRACE is an important plus for MIT.

MIT SAVEPICT/READPICT COMMANDS:

There is a SAVEPICT command in MIT and not in APPLE. The command saves the graphics on screen to disk as a picture file (as contrasted to a file of procedures) which may be recalled with a READPICT command. We acknowledge that this is a real space saver in more complex procedures, and a real timesaver

MIT TERRAPIN/KRELL VS APPLE LCSİ LOGOS

because complex designs can take a long time to draw.

In our teacher programmed Headings and XCOR/YCOR games we use READPICT to erase and replace the rather complex background screen after each turn. That would take significantly longer in APPLE, where it would have to be redrawn each time. It also takes up a lot of workspace and would limit the sophistication of the games.

We have found SAVEPICT/READPICT advantageous when printing out drawings in the classroom. However our rule is no picture may be printed out unless accompanied by procedures that draw it. This prevents command level drawing followed by SAVEPICT, a maneuver students sometime employ to circumvent the discipline of the editor and notetaking. CONCLUSION: APPLE's lack of SAVEPICT and READPICT is regrettable.

EXTENT OF PRIMITIVE VOCABULARY:

APPLE has a larger number of useful primitive commands. Of particular significance are POS and SETPOS for saving as variables key screen positions and returning to them later in the procedure. We acknowledge it is simple to write derived procedures for these in MIT, but they tend to clutter up the workspace. If there were a 'BURY' command in MIT this wouldn't be as significant. Other useful APPLE commands that don't exist as MIT primitives are WAIT, and DOT. There are additional good primitive commands exclusive to APPLE, but we don't see them as particularly significant at the grade levels concerned. CONCLUSION: APPLE'S primitive vocabulary is superior for grades 2 thru 4.

THE APPLE 'PACKAGE' COMMAND:

The PACKAGE command allows a group of related procedures to be selected from the entire workspace and saved under a chosen file name. Because the workspaces of student programmers tend to get VERY cluttered, this command might appear to be useful. However it may require a degree of meticulousness seldom encountered at the grade levels at which we work, and may lead to much down time helping a student find or reprogram one or more procedures he or she left out of a 'package'. CONCLUSION: Of doubtful value.

INSTRUCTION MANUALS:

APPLE has the reputation of having an excellent manual, but this is irrelevant at our grade level. It has not seemed to

us that children at our grade levels make much use of manuals. The TERRAPIN TUTORIAL is equally good for MIT LOGO. Thus the comparison is only of benefit to teachers. CONCLUSION: A draw.

THE APPLE 'BURY' COMMAND:

This enables the writing of aid-for-beginners procedures to be prepared by the teacher and buried, that is to say, they are in the workspace ready for use but cannot be edited, changed, or viewed by the students. CONCLUSION: Useful, minimizes clutter in the visible workspace.

TYPING EASE AND SYNTAX SIMPLICITY:

This we consider an important point which was primarily responsible for our election to use MIT LOGO for the latter half of this year. Because you don't need to quote the names of procedures, because of the better editor and the simple typing of 'ED' to get back to the most recently edited procedure, because of a somewhat simpler syntax (e.g., no brackets needed in parts of IF, IFTRUE, IFFALSE, and TEST commands), because a REMARK in a procedure need not be enclosed in brackets, and because you don't need to ERASEFILE a file before you SAVE a file of the same name, MIT LOGO is easier for children to type and otherwise handle. CONCLUSION: Important advantages for MIT, --- however, see following.

AUTOMATIC RETYPING:

Children's typing skills are a hindrance. Any feature simplifying this aspect of LOGO is welcome. The command 'CONTROL P' in MIT will retype the last command returned ONLY IF a REPEAT command was NOT included in that last command, a most severe limitation. 'CONTROL P' is also nullified if a READ or SAVE command is used (a nuisance with frequent misspellings, or if a disk error has occurred). By contrast, APPLE'S command 'CONTROL Y' is far more powerful as a typing aid. It not only will retype the last command returned under almost any circumstances, but it will repeatedly retype whatever was erased by the command 'CONTROL K', often used to "kill" entire lines in a procedure. Such a line goes into a "kill buffer" and may be reinserted at another location in the procedure with 'CONTROL Y'. This greatly facilitates reordering of command sequence in procedures. CONCLUSION: Retyping with 'CONTROL Y' is a neat and important plus for APPLE.

MIT TERRAPIN/KRELL VS APPLE LCSII LOGOS

SPEED OF DRAWING:

The turtle moves faster in APPLE. This is a bit of a time saver, particularly in the absense of a SAVEPICT command to redraw complex designs. CONCLUSION: The edge to APPLE.

SAVING FILES:

At the end of a class, students must save the procedures they have been working on so as not to lose their work and to be able to start next class where they left off. In MIT LOGO, you simply SAVE a filename, and if that filename is already in the disk catalog, the current workspace is saved over and obliterates the previous filename contents. In APPLE, it is necessary to ERASEFILE a previous filename before you can save under that filename. We consider this an unnecessary addition to typing complexity particularly at the often chaotic time when one class is ending and another beginning. CONCLUSION: Another plus for MIT syntax simplicity and typing ease.

REMARK CAPABILITY:

Adding remarks to procedure lines we believe to be good programming, obliges students to explain matters, and is certainly helpful in reminding children where they are in a procedure, or why a certain line is there. The procedures, which always accompany the printouts, can, with appropriate remarks, be more easily be explained with pride to others. In MIT LOGO, this is accomplished with a simple semicolon, after which anything you write doesn't affect the procedure. In APPLE, the remark must be enclosed in square brackets as a function of TO REM :REMARK -- END. CONCLUSION: Another modest typing simplicity advantage to MIT.

CLEANING UP A STUDENTS CLUTTERED WORKSPACE:

Student programmers frequently read in or load files on top of files, then save and re-read the results back into the workspace. The result is a very long POTS list and a cluttered workspace. The teacher needs to be able to clean up such a workspace quickly with minimum risk of wiping out any procedure of current interest to the student. The following neat method can be done in MIT LOGO:

ED ALL puts all the procedures in the workspace on the edit screen. CONTROL G leaves the editor, but with all the procedures IN THE EDIT BUFFER. ER ALL erases the entire workspace, but not the edit buffer. ED brings back the

MIT TERRAPIN/KRELL VS APPLE LCSI LOGOS

contents of the edit buffer into the editor. At this point we can use CONTROL K, the Repeat key, and up/down cursor moving commands to riffle through the editor erasing whatever we want while being able to see the contents of every procedure and to decide its relevance to the student's current project. In APPLE LOGO, ED ALL procedures must be added (and perhaps buried), and if carefully done, (see attachment), it works in a manner very similar to the MIT method as described above. It is, however, tricky to prepare. CONCLUSION: A modest plus for MIT in workspace cleanup.

SUMMARY

Obviously the decision to use one dialect over another has been for us a subtle judgement based on our own weighting and intuition rather than any obvious overriding feature. We emphasize applicability to grades 2 thru 4. We would predict less weight in future to MIT's TYPING EASE, more weight to MIT's TRACE and SAVEPICT/READPICT commands, and more weight to APPLE's more extensive primitive vocabulary.

Leaving out the minor advantages and draws, we are left with the following criteria. The number of +'s indicates weighting

	APPLE	MIT
EDITING:		++
SCREEN MODES: (for WINDOW)	+++	
DEBUGGING: (for TRACE)		+++
EXTENT OF PRIMITIVE VOCABULARY:	++	
APPLE "BURY" COMMAND:	+	
TYPING EASE AND SYNTAX SIMPLICITY:		+++
SAVING FILES:		+
AUTOMATIC RETYPE: (for CONTROL Y)	+++	
SPEED OF DRAWING:	+	
REMARK FACILITY:		+
WORKSPACE CLEANUP FACILITY:		+
SAVING PICTURES (SAVEPICT/READPICT):		++

Reinhold D. Wappler
252 Carter Street
New Canaan, Conn. 06840

July 2, 1983

AN 'ED ALL' CAPABILITY IN APPLE LOGO:

The procedures published in TURTLE NEWS to allow an ED ALL command in APPLE LOGO to edit the entire workspace are as follows:

```
TO ALL1 :LIST
  IF FIRST :LIST = "PENDOWN [OUTPUT []]
  IF DEFINEDP FIRST :LIST [OP SE FIRST :LIST
    ALL1 BF :LIST] [OP ALL1 BF :LIST]
END

TO ALL
  OUTPUT ALL1 .CONTENTS
END
```

In APPLE LOGO, all names of procedures and names of variables, buried or not, become appended to the front of the .CONTENTS list. When the APPLE LOGO language disk is booted, the .CONTENTS list contains the name of every primitive in the language, beginning with the command PENDOWN. As new procedures and/or variables are created, their names are added to the front of the .CONTENTS list. The two procedures above are equivalent to typing ED followed by square brackets enclosing every name in the .CONTENTS list that has the property of being defined (DEFINEDP), up to but not including the name PENDOWN.

A valuable use of ED ALL that is worth mentioning is to meet the need to clean up a workspace, particularly a student's workspace. Files have been loaded on top of files and the POTS list is a couple of screensfull long. In MIT LOGO, one types this sequence of primitive commands: ED ALL, CONTROL G, ER ALL, and ED. These commands erase the workspace but leave the edit buffer on the screen with all procedures intact that were in the workspace. With a full view of the contents of all procedures, you may then "kill" all unwanted procedures with the CONTROL K command, using the REPEAT key (carefully) to hasten things along. CONTROL C then redefines whatever is left. The above procedures (which may be buried in a startup file) add this neat cleanup facility to APPLE LOGO.

There is a small problem with the ALL and ALL1 procedures

above as they are written. The command ED ALL will include and display in the editor ALL procedures in the workspace, including any that are buried, for example, in a startup file. If for the sake of an uncluttered workspace for a young programmer you wish to exclude procedures which have been buried, such as those in the STARTUP file on the APPLE LOGO language disk, you must modify the procedure ALL1 to include an additional condition. You may rely on the fact that all procedures in the STARTUP file have attached to them the property PROCPKG with the value AIDS. ALL1 is adjusted to select for output only those names in the .CONTENTS list which have both the property of being DEFINED, and also the property of NOT having a PLIST (property list) equal to [PROCPKG AIDS]. Thus:

```
TO ALL1 :LIST
  IF FIRST :LIST = "PENDOWN [OP []]
  TEST AND (DEFINEDP FIRST :LIST) (NOT PLIST FIRST :LIST =
  [PROCPKG AIDS])
  IFTRUE [OP SE FIRST :LIST ALL1 BF :LIST]
  IFFALSE [OP ALL1 BF :LIST]
END
```

To add ALL and ALL1 to the APPLE LOGO language disk STARTUP file, so that they too will be excluded from the procedures displayed in the editor following the ED ALL command, you must assign them the property list PROCPKG with the value AIDS. Do the following:

- 1) Boot the language disk and press return to load the STARTUP file.
- 2) Type in the ALL and ALL1 procedures.
- 3) Type PACKAGE "AIDS [ALL ALL1].
- 4) Type BURY "AIDS.
- 5) Type ERASEFILE "STARTUP
- 6) Type SAVE "STARTUP "AIDS.

The command PACKAGE (3) gives ALL and ALL1 the property "PROCPKG with the value "AIDS. The command BURY (4) gives ALL and ALL1 the property "BURY with the value "TRUE. ALL and ALL1 are now conformed in terms of property lists with the procedures already in the STARTUP file, and will not be selected for output by ED ALL.

It remains only to erase (5) the STARTUP file from the APPLE LOGO language disk and resave (6) in the file STARTUP all procedures having the property "AIDS.

Feb 12, 1984

UPDATE

Version 2 of Terrapin Logo has been marketed since the publication of the comparison of Apple (LCSI) and MIT dialects in May of 1983. The following changes in Terrapin version bear directly on our conclusions.

1) EDITING:

a) The addition of the command CTRL X to kill a line in the editor, and the ability to repeatedly yank back the deleted line with CTRL Y has further improved the editing in version 2.

b) The use of the up and down arrows on the Apple IIe to move the cursor up lines and down lines in the editor is a neat and unique feature.

c) In Version 2 Terrapin Logo it is now possible to call up into the editor any number of related procedures by typing ED PROC1 PROC2 J.

2) AUTOMATIC RETYPE:

a) Apple's advantage in the CTRL Y retype feature is cancelled by the addition of this feature in Terrapin's Version 2.

3) EXTENT OF PRIMITIVE VOCABULARY:

a) Terrapin Version 2 has added COUNT, ITEM, MEMBER?, EMPTY?, and LOCAL. Apple's advantage in this area is reduced substantially, but we still miss DOT, POS, SETPOS, and WINDOW.

4) SAVING FILES:

a) Terrapin Version 2 now allows selected sets of procedures to be SAVED in a filename (SAVE "MYFILE [PROC1, PROC2, ...]). This is a reasonable equivalent to Apple's PACKAGE command.

REINHOLD D. WAPPLER
252 CARTER STREET
NEW CANAAN, CONN. 06840